

全商情報処理検定試験 1 級プログラミング部門 第 73 回検定解説

【1】 解答以外の解答群の語句の説明は以下のとおりである。

- ア. WebサーバとWebブラウザとの間で、HTML文書や関連した画像などのデータを送受信するためのプロトコル。通信が暗号化されていない。
- ウ. 浮動小数点演算で絶対値がほぼ等しい二つの数の差を求めた際、有効数字の桁数が極端に少なくなる現象。
- エ. コンピュータシステムを評価する指標の一つで、誤操作や過負荷、障害などによるデータの破壊や消失がなく、また、改ざんされていないなど、一貫性が保たれていることを表したもの。
- オ. TCP/IP ネットワークで標準的に用いられ、電子メールを送信する際や、メールサーバ間で電子メールを転送する際に用いられるプロトコル。
- ク. サーバなどに接続された日付と時刻、接続元のIPアドレスやドメイン名、ファイル名などコンピュータの接続履歴を時系列で記録したもの。
- ケ. オブジェクト指向において、オブジェクトが持つ性質を定義したクラスに対して具体的な属性値を与えることにより、メモリ上に実体として生成されるもの。
- サ. 除算などの計算処理の際、表現可能な桁数より小さな値を端数処理したことが原因で生じる誤差。

【2】 解答以外のB群の説明文は、以下の語句についての説明である。

- イ. インシデント
- エ. 機能テスト
- オ. フォールトアボイダンス
- キ. ネットワークアドレス
- コ. 外部設計

【3】

1. **方法①** 2進数のまま加算し、その後10進数に変換する。

$$\begin{array}{r} 1 \ . \ 1 \ 0 \ 1 \\ + 1 \ . \ 1 \ 1 \ 1 \\ \hline 1 \ 1 \ . \ 1 \ 0 \ 0 \end{array}$$

(11.1)₂→(3.5)₁₀ そのため、結果はイ. 3.5である。

方法② それぞれ10進数に変換後、加算する。

$$(1.101)_2 \rightarrow (1.625)_{10}$$

$$(1.111)_2 \rightarrow (1.875)_{10}$$

そのため、 $1.625 + 1.875 = 3.5$ となり、結果はイ. 3.5である。

2. 解答以外の語句の説明は以下のとおりである。

ア. インターネットにおける通信の暗号化方式として広く知られ、通信相手の認証やデータの改ざん検知などを行うプロトコル。SSLの次世代規格がTLSであり、SSL/TLSと表記されることもある。

イ. 広く一般に公開され、暗号化することができる公開鍵と、受信者のみが管理する秘密鍵を使用した暗号方式。受信者は、送信者が増えても一つの秘密鍵を管理すればよいので負担を軽減することができる。

3. 解答以外の語句の説明は以下のとおりである。

ア. 実行中に自分自身を呼び出しても、正しく実行することができるプログラムの性質（再帰）。

イ. 複数のタスクが同時に共有して実行しても、正しく実行することができるプログラムの性質（再配置可能）。

4. 解答以外の語句の説明は以下のとおりである。

イ. ASCIIコード以外の画像や音声などのデータを、電子メールで送受信するための規格。

ウ. 音声データをパケット化し、インターネット回線を利用して音声データを送受信する技術。

5. 計算式は次のとおり

$$\frac{6GB}{16Gbps \times 0.75} = 4 \rightarrow \underline{4秒}$$

Point! バイトとビットの単位を合わせ、次のように計算するとよい。

$$\begin{aligned} \frac{6GB \times 8(\text{ビット})}{16Gbps \times 0.75} &= \frac{6GB \times 8(\text{ビット})}{162Gbps \times 0.75} \\ &= \frac{6}{2 \times 0.75} (\text{秒}) \\ &= \frac{6}{1.5} \\ &= 4 \end{aligned}$$

【4】

問1 インサージョンソート（降順）

(1) **比較元**の添字 g の動き。 g は 2 番目から始まり n 番目までのため「 $g = 2$ To n 」となる。

考え方① 下の処理「Fish(0) = Fish(g)」から g が Fish(0) に転記される比較元の添字であることに気づく。また、その下の比較先の動きである For 文で「 $j = g - 1$ To Step -1」より「 **$g = 2$** To n 」になることに後で気づく（考え方③）。

考え方② (2) の下の処理で「Fish($j + 1$) = Fish(j)」と Fish(j) の値を後ろに入れる（ずらす）ことから、インサージョンソートであることに気づく。

考え方③ 考え方①②より、**比較元**は 2 番目から n 番まで**範囲を広げて**動くこと「 $g = 2$ To **n** 」に気づく。

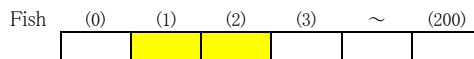
(2) Fish(0) と比較する処理。降順であるため「**Fish(j) < Fish(0)**」のときに、下の処理で**比較先の Fish(j) の値**を後ろに入れる（ずらす）。なお、上の For 文は**比較先の添字 j の動き**であり、 j は**比較元の 1 つ前から**になるため、「 $j = \underline{g - 1}$ 」となる。

For (1) Fish(0) = Fish(**g**)
 For j = g - 1 To 1 Step -1
 If (2) Then Fish(j + 1) = Fish(j)
 Else Exit For
 End If
 Next j
 (3)
 Next g

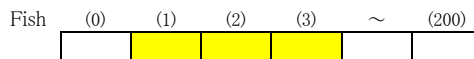
例：トレース表（簡略） $n = 200$ の例

g	j		
2	1	※1	
3	2	※2	
	1		※3
			200
			199
			198
			:
			1

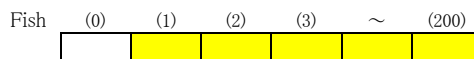
※1 $g = 2$ のとき・・・1~2 番までが範囲内で降順になる



※2 $g = 3$ のとき・・・1~3 番までが範囲内で降順になる



※3 $g = 200$ のとき・・・1~200 番までが範囲内で降順になる



(3) Fish(0) を範囲内での正しい場所 (Fish(j) が Fish(0) 以上だったため、Fish($j + 1$)) に挿入する。

Point! (1) と (2) の上の For 文の条件はセットで考えてください。
 二重ループのこと

問2 二分探索 (昇順) ・リカーシブ (再帰) 呼び出し

- (4) データが 見つからなかったか 判定する。上の処理で戻り値が Kekka に代入されるのでデータが見つからなかった場合は、上の処理で Kekka に -1 が代入されるため、「Kekka = -1」となる。
- (5) 中央値の指すデータ Cod(Tyu) と Dat の比較。下の処理が「Tansaku = Tansaku(Cod, Dat, Tyu + 1, Jogen)」のため、中央値に + 1 する 下限の更新 であることに気づき「Cod(Tyu) < Dat」となる。

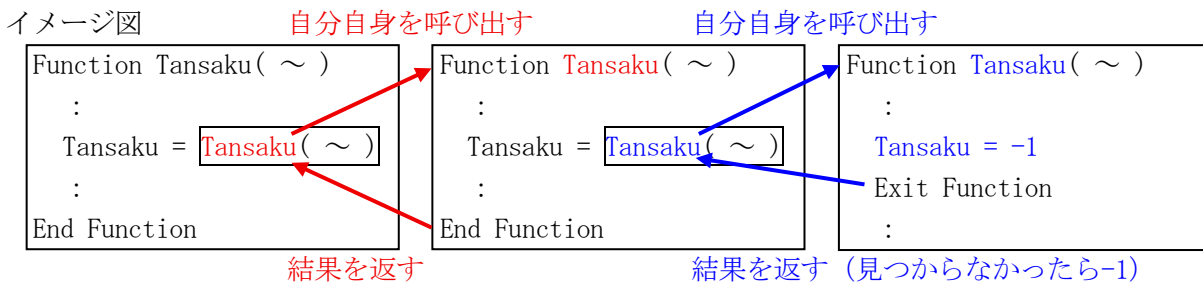
Point! 引数の対応を見つけてください。

```

Sub Program2(Code() As Long, n As Long)
    Kekka = Tansaku(Code, Data, 0, n - 1)
    : データが入った配列 入力した数値 配列の下限 配列の上限
    : 引数を渡した Function プロシージャから帰ってきた値(1 か -1) を Kekka に入れる
End Sub

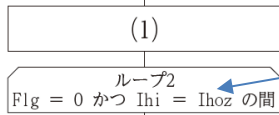
Function Tansaku(Cod() As Long, Dat As Long, Kagen As Long, Jogen As Long) As Long
    : データが入った配列 探す数値 探索の下限 探索の上限
    :
    : Tansaku = -1 データが見つからなかった場合 Tansaku に -1 を入れる
    Exit Function Function プロシージャを直ちに終了する
    :
    : Tansaku = 1 データが見つかった場合、Tansaku に 1 を入れる
    :
    : Tansaku = Tansaku(Cod, Dat, Tyu + 1, Jogen) 下限を更新する場合
    : 探索の下限を中央値 + 1 にして新たに Tansaku へ渡す (再帰)
    :
    : Tansaku = Tansaku(Cod, Dat, Kagen, Tyu - 1) 上限を更新する場合
    : 探索の上限を中央値 - 1 にして新たに Tansaku へ渡す (再帰)
    :
End Function Tansaku の値を返して Function を終わる
    
```

Point! 再帰呼び出しを確認しましょう。

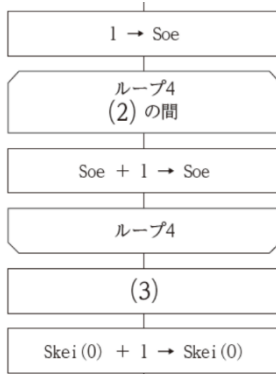


【5】グループトータル・線形探索・二次元配列

- (1) グループのキーとなる Ihi を Ihoz に保存する。ループ2の繰り返し条件「 $Ihi = Ihoz$ 」およびループ3の繰り返し条件「 $Kban = Khoz$ 」とその上の処理「 $Kban \rightarrow Khoz$ 」をヒントにする。



- (2) 線形探索。「受付時刻をもとに」線形探索をするため、配列の値が昇順に記憶されていることから受付時刻が配列の値より大きい間繰り返す「 $Uji > Kubun(Soe)$ 」となる。



例: Uji が 1030 (10:30) の場合 ○ は比較先の値

Soe = 1 のとき

1030 > 1029 のため Soe + 1 → Soe を実行し Soe が 2 になる

Kubun	(0)	(1)	(2)	~	(12)	(13)	(14)
		1029	1059	~	1559	1629	1659
		(10:00~)	(10:30~)	~	(15:30~)	(16:00~)	(16:30~)

Soe = 2 のとき

1030 < 1059 となるため Soe は 2 のままループ4を抜ける

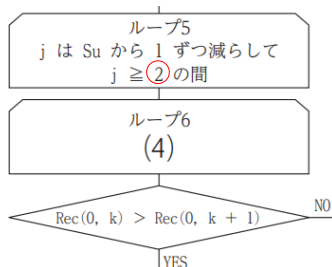
Kubun	(0)	(1)	(2)	~	(12)	(13)	(14)
		1029	1059	~	1559	1629	1659
		(10:00~)	(10:30~)	~	(15:30~)	(16:00~)	(16:30~)

そのため、(3) で Skei(2) + 1 → Skei(2) として集計される

- (3) 配列 Skei に区分ごとの参加者数を集計 (+1) する。
 (4) 保存されている企業番号 (Khoz) より企業名を表示する。
 (5) 配列 Gkei の 1 行目に区分ごとの合計を求める。日計を (5) で配列 Gkei の 1 行目 である合計に集計している。ループ5の中の処理「 $Gkei(0, f) + Skei(f) \rightarrow Gkei(0, f)$ 」は、日計を集計する処理であるので参考にする。

【6】線形探索・最小値の検出・二次元配列・バブルソート

- (1) 部員数のカウント。ループ3の繰り返し条件の終了値「 $g \leq Su$ 」および、ループ5の繰り返し条件「 j は Su から・・・」およびループ8の終了条件「 $p \leq Su$ 」をヒントにし、 Su が部員数であることに気づくこと。
 (2) 配列 Rec0 の行目、最速記録 (週) の初期値 (仮の最小値) の設定。仮の最小値として $Rec(1, g)$ を $Rec(0, g)$ に保存する。ループ4で最小値の検出に使う添字 h が 2 行目 (火曜日) から始めていることから気づくこと。
 補足 最小値の検出には初期値に「99.99」を用いることが多いが、この方法のほうが効率的である。
 (3) 最小値 (最速記録 (週)) の検出。下の処理「 $Rec(h, g) \rightarrow Rec(0, g)$ 」で転記 (最小値の更新) していることから気づくこと。
 (4) バブルソートの比較元の添字 k の繰り返し条件。ループ5でソートの1回ごと最後の比較元の列番号である j の値で、繰り返し条件の終了値が「 $j \geq 2$ 」であることから (4) の終了値「 k は 1 から・・・ $k \leq j - 1$ 」となる。つまり $j = 2$ のときにループ6の繰り返し条件が「 k は 1 から・・・ $k \leq 1$ 」になり、次の判定処理でも「 $Rec(0, k) > Rec(0, k + 1)$ 」であることから最後の判定は「 $Rec(0, 1) > Rec(0, 2)$ 」になることから気づくこと。



(4) k は 1 から 1 ずつ増やして $k \leq j - 1$ の間

- (5) 備考に表示する値を判定するための多分岐。配列 Rec の 0 行目 (最速記録 (週)) を用いる。

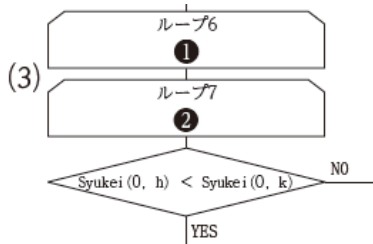
【7】二次元配列・順位付け（降順・どちらか小さいほうに+1（逓減比較法））・セレクションソート（降順）・順位付け（降順・整列済みのデータを用いる）・添字の入れ子

処理の概要

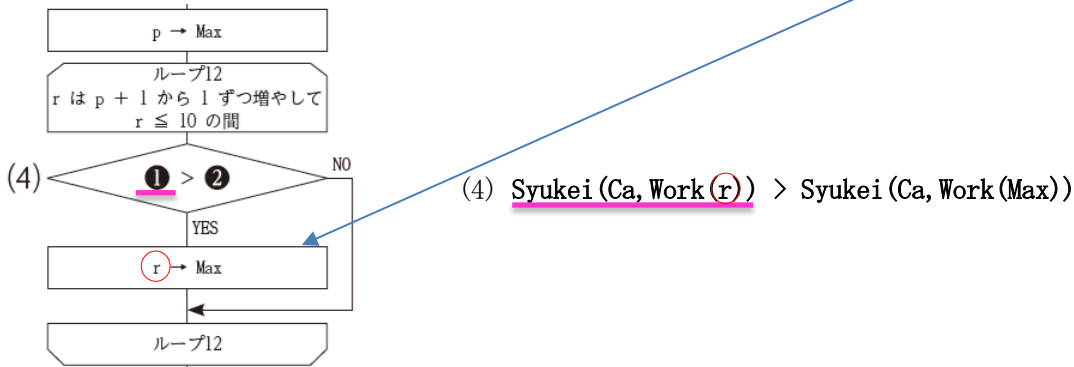
- | | |
|---|--|
| 1 | 配列 Syukei にカテゴリ番号, 店舗コードごとに売上金額を集計←ループ 2 |
| 2 | 配列 Juni に順位を求める←ループ 4~7 |
| 3 | 配列 Work を利用し, カテゴリ番号の売上金額計の降順に並べ替える←ループ 9~12 |
| 4 | カテゴリにおける店舗ごとの売上金額計に順位を求める。←ループ 13 |

空欄の解説

- (1) 売上金額の計算。次の処理で Kin を加算していることをヒントにする。
- (2) 順位の初期化。ループ 4 繰り返し条件の終了値「 $f \leq 2$ 」より, f が行方向の添字であることに気づく。
- (3) 順位付けの処理。売上金額, 客数計, 客単価の 3 つを降順に順位付けする。下の処理「 $Jun(0, h) + 1 \rightarrow Jun(0, h)$ 」および「 $Jun(0, k) + 1 \rightarrow Jun(0, k)$ 」により, どちらか小さいほうに+1していることに気づく（逓減比較法）。そのため 10 店舗であることから ①「 h は 1 から $\cdot \cdot h \leq 9$ 」, ②「 k は $h+1$ から $\cdot \cdot k \leq 10$ 」の組み合わせとなる。



- (4) 売上金額計の仮の最大値との比較（最大値の検出）。(4) の判定が真のときに次の処理で Max を r で更新「 $r \rightarrow Max$ 」していることから, 入力されたカテゴリで比較先の添字 r が指す $Work(r)$ に記憶されている店舗コードの売上金額計 ($Syukei(Ca, Work(r))$) がこの時点での仮の最大値となる。そのため ①「 $Syukei(Ca, Work(r))$ 」, ②「 $Syukei(Ca, Work(Max))$ 」の組み合わせとなる。なお, 店舗コードは配列 Work に保存されている値を用いている（添字の入れ子）ので注意が必要である。



- (5) 順位・店舗名・売上金額計の表示。上記 (4) と同じく, 店舗名や売上金額計の表示に用いる店舗コードは配列 Work に保存されている値を用いているので注意が必要である。

Point! この問題では、ループ10で配列Workの初期値を設定し、その後の処理でWorkを使っているので注意する。

例：（本問題と同じデータではありません）

初期値設定後

Work	(0)	(1)	(2)	(3)	~	(10)
	交換用	1	2	3	~	10

ループ11・12 例:p = 1, Max = 3で並べ替えが発生した場合

Work	(0)	(1)	(2)	(3)	~	(10)
	交換用	1	2	3	~	10

交換処理

Work(p) → Work(0)

Work	(0)	(1)	(2)	(3)	~	(10)
	1	1	2	3	~	10

Work(Max) → Work(p)

Work	(0)	(1)	(2)	(3)	~	(10)
	1	3	2	3	~	10

Work(0) → Work(Max)

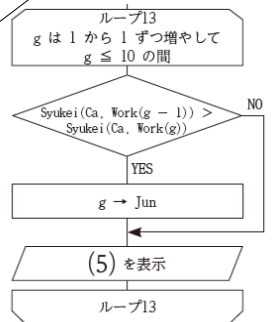
Work	(0)	(1)	(2)	(3)	~	(10)
	1	3	2	3	~	10

ループ13（並べ替え後）例:g = 10

Work	(0)	(1)	(2)	~	(9)	(10)
	?	1	6	~	?	9

Tmei	(0)	(1)	(2)	~	(9)	(10)
		A店	B店	~	I店	J店

ループ13でここに記憶されている店舗コード順に10件出力される。



この例ではループ13でCa = 8, g = 10のとき、Tmei(Work(g))ではWork(10)に9が記憶されているため、配列Tmei(9)により店舗名「I店」を表示する。

カ. Jun, "位", Tmei(Work(g)), Syukei(Ca, Work(g))

10 I店 72,500

実行結果

(売上分析)						
(店舗名)	(売上金額合計)	(順位)	(客数計)	(順位)	(客単価)	(順位)
A店	958,200	6位	60	4位	15,970	8位
J店	998,400	4位	60	4位	16,640	5位
(分析したいカテゴリ番号を入力) 8						
(カテゴリ名) マグカップ						
(順位)	(店舗名)	(売上金額計)				
1位	A店	150,100				
2位	F店	125,000				
2位	G店	125,000				
10位	I店	72,500				

(第3図)